Progress In Electromagnetics Research, Vol. 120, 165–179, 2011

AN ACCURATE CONFORMAL FOURIER TRANSFORM METHOD FOR 2D DISCONTINUOUS FUNCTIONS

C. H. Zhu[1], Q. H. Liu[2, *], Y. H. Liu[3], Y. Shen[1], and L. J. Liu[1]

[1]Department of Control Science and Engineering, Harbin Institute of Technology, 92 West Street, Harbin 150001, China

[2]Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708, USA

[3]Department of Electronic Science, Xiamen University, Fujian 361005, China

Abstract—Fourier transform of discontinuous functions are often encountered in computational electromagnetics. A highly accurate, fast conformal Fourier transform (CFT) algorithm is proposed to evaluate the finite Fourier transform of 2D discontinuous functions. A curved triangular mesh combined with curvilinear coordinate transformation is adopted to flexibly model an arbitrary shape of the discontinuity boundary. This enables us to take full advantages of high order interpolation and Gaussian quadrature methods to achieve highly accurate Fourier integration results with a low sampling density and small computation time. The complexity of the proposed algorithm is similar to the traditional 2D fast Fourier transform algorithm, but with orders of magnitude higher accuracy. Numerical examples illustrate the excellent performance of the proposed CFT method.

1. INTRODUCTION

Fourier transform (FT), as a most important tool for spectral analyses, is often encountered in electromagnetics, such as scattering problems [1–4], analysis of antennas [5, 6], far-field patterns [7, 8] and many others [9, 10].

The traditional fast Fourier transform (FFT) algorithm is the most popular approach to evaluate the Fourier transform. In practice,

Received 23 July 2011, Accepted 26 August 2011, Scheduled 3 September 2011

* Corresponding author: Qing Huo Liu (qhliu@ee.duke.edu).

however, many functions to be transformed are discontinuous across the boundary of an irregular area. For example, in volume integral equation solvers in electromagnetics, some components of the unknown electric current density fields to be transformed are discontinuous across the material interfaces, which in general have arbitrary shapes; another example is the analysis of radiation patterns of reflector antennas and planar near-field to far-field transformation, where the Fourier transform integral of spatially limited functions with discontinuities at the boundary has to be evaluated [11, 12]. For this kind of functions, however, there usually exist significant stair-casing errors due to the uniform Cartesian orthogonal grid required by the traditional 2D FFT algorithm, and the accuracy is limited since FFT is based on the trapezoidal quadrature scheme.

Some works have been done to improve the accuracy for one-dimension (1D) piecewise smooth functions [1, 11, 13–17]. Direct extension of these algorithms to high dimensions still requires that the area is meshed into a Cartesian orthogonal grid [11], which is not flexible for an arbitrary boundary shape.

A fast algorithm for evaluating the Fourier transform of 2D discontinuous functions has been proposed in [12] as an extension of [11] to allow an arbitrary boundary shape. This method adopts a triangular mesh to conform with the arbitrary shape of the discontinuity boundary, the Gaussian quadrature to increase the accuracy, and 2D nonuniform fast Fourier transform (NUFFT) [18–21] to provide a computational complexity comparable with the traditional FFT algorithm. However, with only straight triangles, only linear interpolation and second order Gaussian quadrature have been applied. It is natural to expect that a higher order method can achieve higher accuracy, if the mesh can still conform well with the discontinuity boundary. This work is inspired by and realizes this idea. By mapping a curved triangle to a straight one with curvilinear coordinate transformation, it allows to take full advantage of higher order interpolation and integration algorithms without losing the flexibility of conforming with an arbitrary boundary shape. The complexity of the proposed algorithm is $O(\mu^2 N_1 N_2 \log(\mu^2 N_1 N_2) + MQ^2)$, which is similar to the traditional 2D FFT algorithm, where $M = LI$; L is the number of the triangles and I is the number of the quadrature points in each triangle. N_1 and N_2 are the numbers of sampling points in the frequency domain in each dimension, $\mu = 2$ and $Q \ll \max(N_1, N_2)$ is a constant. Numerical results illustrate the advantages of the developed algorithm over the traditional 2D FFT algorithm and the algorithm in [12].

2. FORMULATIONS AND ALGORITHMS

The objective of this work is to develop a fast and accurate algorithm for evaluating $F(u,v)$, the finite Fourier transform of a 2D piecewise smooth function $f(x,y)$,

$$F(u,v)=\int_S f(x,y)e^{j2\pi(ux+vy)}dxdy \tag{1}$$

where S is an area composed of one or multiple regions with arbitrary boundary shape, i.e., $S=\bigcup_{i=1}^{I_s} S_i$ and S_i is a 2D region of arbitrary shape. $f(x,y)$ is assumed to be continuous inside each S_i.

In this section the tools used are first introduced and then the 2D conformal Fourier transform (2D CFT) algorithm is formulated.

2.1. Interpolation over a Triangle

A uniform Cartesian orthogonal gird, which cannot describe very well the boundary shape of an arbitrary area S unless S is a rectangle, is required when using the traditional 2D FFT algorithm to evaluate the integration (1). In order to reduce the stair-casing errors caused by the orthogonal grid in 2D FFT, a curved triangular mesh is used to discretize the area, which conforms with an arbitrary boundary shape.

The M_Ith order Lagrangian interpolation function $f_{M_I}(x,y)$ of $f(x,y)$ in a triangle can be achieved by

$$f_{M_I}(x,y)=\sum_{m_1=0}^{M_I}\sum_{m_2=0}^{M_I-m_1} f(x_{m_1m_2},y_{m_1m_2})B_{m_1m_2}^{M_I}(s,t) \tag{2}$$

where (s,t) is the Barycentric coordinate of (x,y) in a triangle, and $\{(x_{m_1,m_2},y_{m_1,m_2})\}_{m_1=0,m_2=0}^{M_I,M_I}$ are the Lagrangian interpolation positions in the triangle given by

$$(x_{m_1m_2},y_{m_1m_2})=\left(\frac{a_1m_1+a_2m_2+a_3(M_I-m_1-m_2)}{M_I},\right.$$
$$\left.\frac{b_1m_1+b_2m_2+b_3(M_I-m_1-m_2)}{M_I}\right),$$

where (a_k,b_k), $k=1,2,3$ are the vertices of the triangle. Example for $(x_{m_1m_2},y_{m_1m_2})$ in a right triangle when $M_I=3$ is shown in Fig. 1; the Lagrangian interpolation position in an arbitrary triangle can be mapped from that in a right triangle by linear coordinate transform. $B_{m_1,m_2}^{M_I}(s,t)$, the Lagrangian interpolation basis function of order M_I in

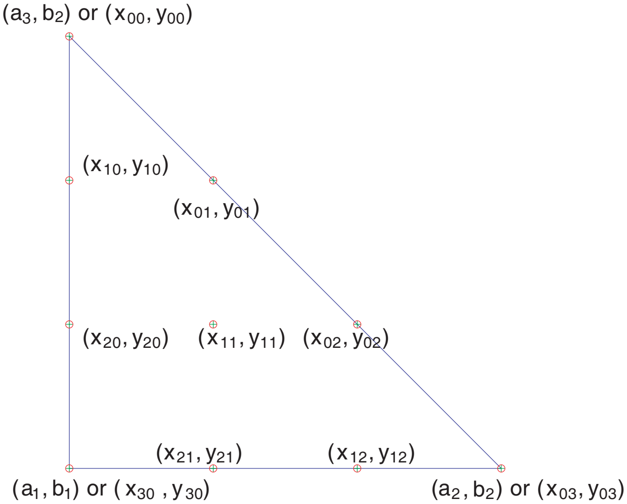


Figure 1. Lagrangian interpolation positions in a right triangle when the interpolation order $M_I = 3$.

the triangle, can be expressed in terms of the Silvester-Ferrari auxiliary polynomial $\{R_m^{M_I}\}_{m=0}^{M_I}$ [22, 23],

$$B_{m_1 m_2}^{M_I}(s,t) = R_{m_1}^{M_I}(s) R_{m_2}^{M_I}(t) R_{(M_I - m_1 - m_2)}^{M_I}(1 - s - t), \tag{3}$$

where

$$R_0^{M_I}(s) = 1,\ R_{m+1}^{M_I}(s) = \frac{1}{m} R_m^{M_I}(s)[Ms - (m-1)], \quad 0 \le s, t \le 1 \tag{4}$$

Note that the Silvester-Ferrari auxiliary polynomial $\{R_m^{M_I}\}_{m=0}^{M_I}$ can be computed recursively by (4).

2.2. Gaussian Quadratures over a Triangle

From the point of view of numerical computation, a spectral analysis algorithm based on the traditional 2D DFT(discrete Fourier transform) algorithm essentially uses a trapezoidal quadrature scheme over a uniform Cartesian orthogonal grid, and the FFT algorithm provides a fast algorithm to compute DFT.

In this work, a curved triangular mesh is used instead to reduce the stair-casing errors. In order to model the triangles with curved edges in the boundary of the area more accurately, curvilinear coordinate transformation is used to map a curved triangle to a straight one;

then the high order symmetrical Gaussian quadrature formulas over the triangles are adopted to evaluate the finite Fourier integration (1) more accurately; finally the 2D NUFFT [20] algorithm helps to make the computational complexity similar to 2D FFT.

2.2.1. Gaussian Quadratures over a Straight Triangle

Denote $\{(s_i, t_i)\}_{i=1}^{I}$ as the Barycentric coordinates and $\{\omega_i\}_{i=1}^{I}$ as the weights for symmetrical Gaussian quadrature over a triangle of order M_g [24]. Denote $Gq(\cdot)$ as the relationship between the number of Gaussian quadrature points I and the order M_g, i.e., $I = Gq(M_g)$. If the area S is meshed into L triangles, the integration (1) can be approximated by

$$\int_{\mathbf{S}} f(x,y)e^{j2\pi(ux+vy)}dxdy \approx \sum_{l=1}^{L} A^l \sum_{i=1}^{I} \omega_i f_{M_I}(g_{1i}^l, g_{2i}^l)e^{j2\pi(ug_{1i}^l+vg_{2i}^l)} \tag{5}$$

where $\{(g_{1i}^l, g_{2i}^l)\}_{i=1}^{I}$ are the symmetrical Gaussian quadrature positions in the lth triangle, whose Barycentric coordinates are $\{(s_i, t_i)\}_{i=1}^{I}$. $A_l = \frac{1}{2}(a_1^l b_2^l - a_2^l b_1^l + a_2^l b_3^l - a_3^l b_2^l + a_3^l b_1^l - a_1^l b_3^l)$ is the area of the lth triangle with vertices $\{(a_l, b_l)\}_{m=1}^{3}$.

2.2.2. A Curved Triangle

For a curved triangle in the mesh, the Gaussian quadrature positions $\{(g_{1i}^l, g_{2i}^l)\}_{i=1}^{I}$ can be achieved by the curvilinear coordinate transformation using the Lagrangian interpolation functions with

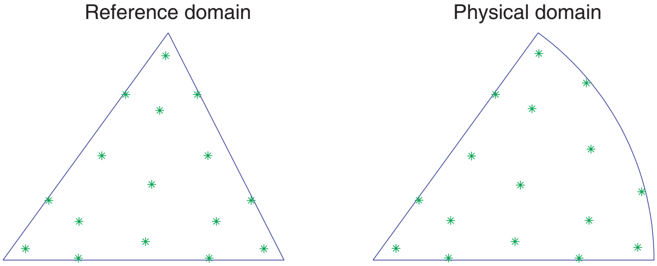


Figure 2. Gaussian quadrature points labeled by "$*$" before and after the curvilinear coordinate transformation for a curved triangle when the curvilinear coordinate transformation order $P = 4$ and the Gaussian quadrature order $M_g = 8$.

simplex coordinates. The Pth order curvilinear coordinate transformation is performed by

$$\begin{cases} g_{1i}^{l} = \sum\limits_{p_1=0}^{P} \sum\limits_{p_2=0}^{P-p_1} x_{p_1p_2} B_{p_1p_2}^{P}(s_i, t_i) \\ g_{2i}^{l} = \sum\limits_{p_1=0}^{P} \sum\limits_{p_2=0}^{P-p_1} y_{p_1p_2} B_{p_1p_2}^{P}(s_i, t_i) \end{cases} \tag{6}$$

where (x_{ij}, y_{ij}) are the Lagrangian interpolation positions in the curved triangle. Here, $B_{p_1,p_2}^{P}(\cdot,\cdot)$ is the basis function of the Pth order Lagrangian interpolation as defined in Equation (3), and $\{(s_i, t_i)\}_{i=1}^{I}$ are the Barycentric coordinates of the symmetrical Gaussian quadrature positions in a triangle. Fig. 2 shows an example of the Gaussian quadrature points before and after the curvilinear coordinate transformation when $P = 4$ and $M_g = 8$. Note that the 2D DFFT method proposed in [12] is actually equivalent to the case with $P = 1$.

2.3. 2D NUFFT

Direct computation of (5) is very time consuming, with the CPU time complexity of $O(LIN_1N_2)$, where N_1 and N_2 are the numbers of discrete points of u and v, respectively, in the frequency domain. Similar to [12], 2D NUFFT [20] is used to reduce the computational complexity. Equation (5) can be rewritten into

$$F(u,v) \approx \sum_{m=0}^{M-1} H_m e^{j2\pi(ux_m+vy_m)} \tag{7}$$

where

$$M = LI,\ H_{(l-1)I+i-1} = A^l \omega_i f_{M_I}\left(g_{1i}^{l}, g_{2i}^{l}\right)$$

and

$$(x_{(l-1)I+i-1}, y_{(l-1)I+i-1}) = \left(g_{1i}^{l}, g_{2i}^{l}\right)$$

Sampling the frequency domain with $(u_{n_1}, v_{n_2}) = (n_1\Delta_1, n_2\Delta_2)$, $n_1 = -\frac{N_1}{2}, \ldots, \frac{N_1}{2} - 1, n_2 = -\frac{N_2}{2}, \ldots, \frac{N_2}{2} - 1$, where (Δ_1, Δ_2) are the frequency resolutions, we obtain

$$F_{n_1,n_2} = \sum_{m=0}^{M-1} H_m e^{j2\pi(n_1x'_m+n_2y'_m)} \tag{8}$$

where $(x'_m, y'_m) = (N_1\Delta_1 x_m, N_2\Delta_2 y_m)$. Equation (8) can be computed by the 2D NUFFT algorithm [20] with complexity of $O(\mu^2 N_1 N_2 \log(\mu^2 N_1 N_2) + M(q+1)^2)$, where usually the oversampling rate $\mu = 2$ and $q \ll N_1, N_2$; note $O(M(q+1)^2)$ is for the pre-NUFFT part, which can be computed only once for the same mesh.

2.4. 2D Conformal Fourier Transform (2D CFT) Algorithm

Now it is ready to summarize the 2D conformal Fourier transform (2D CFT) algorithm with the following steps:

(i) Create the mesh for the support area S with straight triangles inside and curved triangles near the boundary.

(ii) Find the Gaussian quadrature positions $\{(g^l_{1i}, g^l_{2i})\}^I_{i=1}$ and weights $\{\omega_i\}^I_{i=1}$ for triangles in the mesh. For curved triangles, curvilinear coordinate transformation (6) is used.

(iii) Calculate $f_{M_I}(g^l_{1i}, g^l_{2i})$ by the M_Ith order Lagrangian interpolation over triangle as in Equation (2). The complexity is $O(M_I^2 M)$;

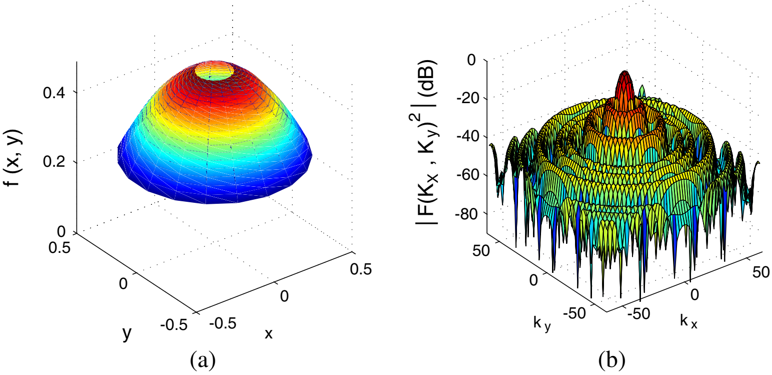
(iv) Apply the 2D NUFFT algorithm to compute Equation (8). The complexity is $O(\mu^2 N_1 N_2 \log(\mu^2 N_1 N_2) + M(q+1)^2)$.

Note that when computing Equation (2) in step (iii), $B^{M_I}_{m_1 m_2}(s,t)$ only needs to be computed once for all straight triangles since the barycentric coordinates of $(g^l_{1i}, g^l_{2i})^I_{i=1}$ are all the same as $(s_i, t_i)^I_{i=1}$. So the dominant computational complexity is $O(\mu^2 N_1 N_2 \log(\mu^2 N_1 N_2) + MQ^2)$, which is similar to that of the traditional 2D FFT algorithm $O(N_1 N_2 \log(N_1 N_2))$, where $Q = \max(M_I, q+1)$. The computation complexity for directly computing of Equation (7) is $O(MN_1 N_2)$, where $M = N_1 N_2$ if the CFT algorithm is preformed under the same sampling density as the traditional FFT algorithm.

Note that the support area S of the spatial function can be composed of one or multiple regions with an arbitrary boundary shape.

3. NUMERICAL RESULTS

In this section, three examples are shown to illustrate the performance of the developed 2D CFT algorithm. Example 1 takes the same function $f(x,y)$ as in [12] so as to compare this algorithm with the 2D FFT algorithm and the 2D DFFT method proposed in [12]; Example 2 shows the results for an oscillatory spatial function; Example 3 compares the 2D CFT algorithm with the traditional 2D FFT algorithm from accuracy, computation time and sampling density required in the spatial domain; Example 4 compares the results by 2D



(a) (b)

Figure 3. The spatial function and its spectra as shown in Equation (9). (a) Spatial function. (b) Spectra.

CFT and 2D FFT algorithm for a spatial function whose support area S includes multiple regions.

Example 1. Consider the spatial domain function as a secondary pattern of an aperture distribution [12]

$$f(x,y)=\begin{cases}0.5-\rho^2, & \rho\in[0.1,0.5]\\ 0, & \text{otherwise}\end{cases} \tag{9}$$

Figure 3 shows the spatial function and its spectra in Equation (9). Define the relative L_2 error as

$$E_2=\frac{|\hat{F}(u,v)-F(u,v)|_2}{|F(u,v)|_2}$$

where $|\cdot|_2$ represents the second norm, and $F(u,v)$ and $\hat{F}(u,v)$ are, respectively, the exact spectrum of $f(x,y)$ and that obtained by 2D CFT. As mentioned in Section 2.2.2, the DFFT-II in [12] is just a special case of 2D CFT with the curvilinear coordinate transformation order $P=1$. To compare our method with that in [12], the same interpolation order $M_I=1$ and Gaussian quadrature order $M_g=2$ are chosen, and the only difference is that $P=2$ is used in this work. Fig. 4 shows the errors versus the sampling density in terms of the number of points per wavelength of 2D CFT, DFFT-II and 2D FFT. It is shown that to achieve 0.1% error, it requires about 3 PPWs (points per wavelength) with 2D CFT, while 6 PPWS with DFFT-II and more than 20 PPWs with 2D FFT.

Example 2. Consider an oscillatory spatial domain function

$$f(x,y)=\begin{cases}J_1(16\pi\rho)e^{j\phi}, & \rho\in[0.1,0.5]\\ 0, & \text{otherwise}\end{cases} \tag{10}$$

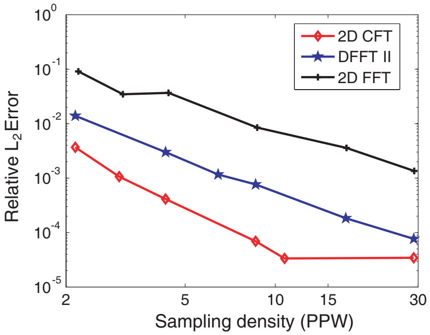


Figure 4. Relative L_2 errors versus the sampling density in terms of the number of points per wavelength (PPW) for 2D FFT, DFFT-II and 2D CFT when $P = 2$.

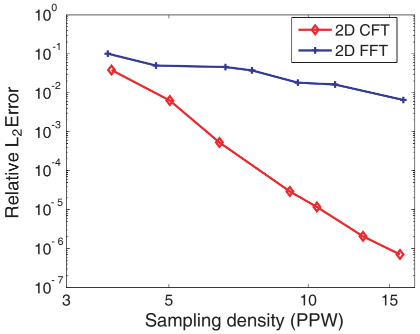


Figure 5. Relative L_2 errors versus the sampling density for 2D FFT and 2D CFT when $M_I = 5$, $M_g = 10$, $P = 6$.

where (ρ, ϕ) are the polar coordinates of spatial domain. Fig. 5 shows the errors versus the sampling density for 2D FFT and 2D CFT when $M_I = 5$, $M_g = 10$, $P = 6$. It is observed that 2D CFT can obtain about four orders of magnitude higher accuracy than 2D FFT with a similar sampling density; for example, CFT can achieve 7.07×10^{-7} relative error at 15.80 PPW, while the relative error with 2D FFT is 6.55×10^{-3} at 16.06 PPW.

Example 3 Consider the spatial domain function with even higher spatial frequency

$$f(x,y) = \begin{cases} J_5(16\pi\rho)e^{j5\phi}, & \rho \in [0.1, 0.5] \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

Figure 6 shows the spatial function $f(x, y)$, whose spectra are shown in Fig. 7. Table 1 shows the comparisons of accuracy, computation time and sampling density in the spatial domain between 2D FFT and 2D CFT, where N_s is the number of sampling points in spatial domain. It shows that for the largest case, the 2D CFT is more accurate and has a speedup factor of 270 over the 2D FFT algorithm.

Example 4 Consider an oscillatory spatial domain function defined in multiple regions

$$f(x,y) = \begin{cases} J_3(16\pi\rho)e^{j3\phi}, & \rho \in [0, 0.4] \cup [0.7, 1] \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

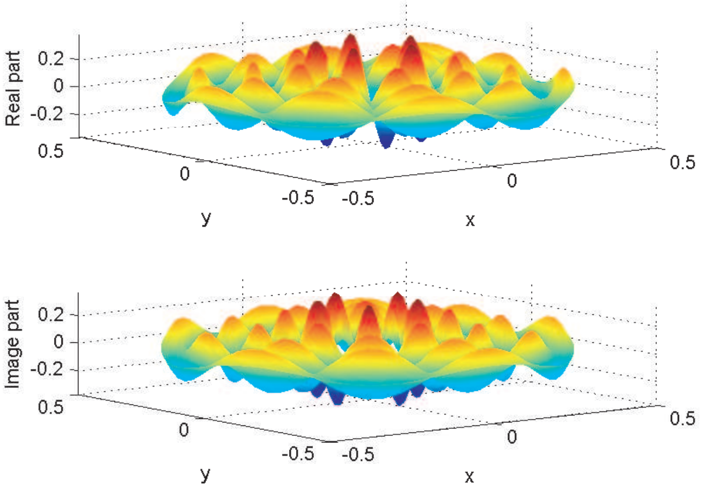


Figure 6. The real and image parts of spatial function $f(x, y)$ in Equation (11).

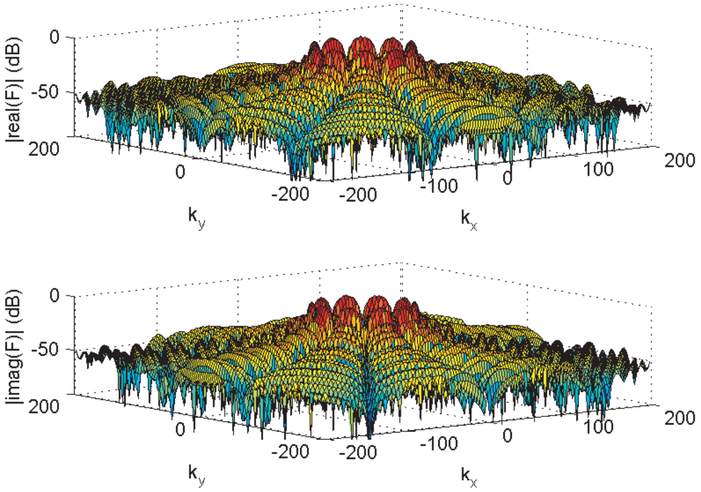


Figure 7. The real and image parts of the spectra for $f(x, y)$ in Equation (11).

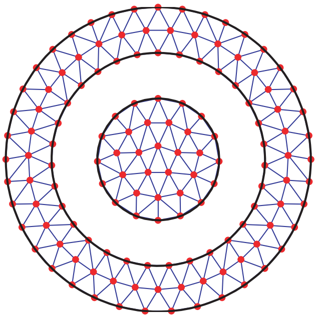


Figure 8. The support area for the spatial function in Equation (12) under a triangular mesh.

Table 1. Comparison between 2D CFT and 2D FFT for Example 3.

	Error(%)	Timing (s)				N_s
2D FFT	0.1220	0.9913				544644
	0.0168	2.8737				2178576
	0.0052	45.4559				34810000
2D CFT		Interp	pre-NUFFT	NUFFT	Total	
	0.1030	0.0268	0.0283	0.0127	0.0679	5640
	0.0156	0.0351	0.0419	0.0196	0.0966	9735
	0.0014	0.0569	0.0675	0.0440	0.1685	18330

Table 2. Comparison between 2D CFT and 2D FFT for Example 4.

	Error(%)	Timing (s)				N_s
2D FFT	0.1607	1.9674				893025
	0.0298	10.4475				8191044
	0.0023	159.9274				130942249
2D CFT		Interp	pre-NUFFT	NUFFT	Total	
	0.1319	0.0161	0.0262	0.0135	0.0558	10017
	0.0281	0.0209	0.0382	0.0177	0.0768	12069
	0.0024	0.0474	0.0931	0.0485	0.1889	29297

Figure 8 shows the support area of the spatial function defined by Equation (12) under a triangular mesh. Table 2 shows the comparisons of accuracy, computation time and sampling density in the spatial domain between 2D FFT and 2D CFT. It shows that, it requires only 0.056 seconds and $N_s = 10017$ to achieve 0.13% error with 2D

CFT, while it requires 1.97 seconds and $N_s = 893025$ to achieve 0.16% error with 2D FFT; it requires only 0.077 seconds and $N_s = 12069$ to achieve 0.028% error with 2D CFT, while it requires 10.45 seconds and $N_s = 8191044$ to achieve 0.030% error with 2D FFT; it requires only 0.189 seconds and $N_s = 29297$ to achieve 0.0024% error with 2D CFT, while it requires 159.93 seconds and $N_s = 130942249$ to achieve 0.0023% error with 2D FFT. That is, for this example, to achieve around 0.0024% errors, CFT is about 850 times faster than FFT.

Example 3 and Example 4 are both performed on a computer with the system type of Microsoft Windows XP professional X64, editor 2003 service pack 2. The processor is Intel core 2Quad CPU Q9550 @ 2.83 GHz 3.83 GHz and the memory (RAM) is 8.00 GB.

4. CONCLUSION

This work develops a high accuracy, fast algorithm to evaluate the Fourier transform of a 2D discontinuous function whose support area has an arbitrary boundary shape. By combining the curved triangle mesh and curvilinear coordinate transformation, it is able to conform with the area shape very well when higher order interpolation and higher order quadrature techniques are applied. Therefore, highly accurate results are achieved through the proposed algorithm. Furthermore, the 2D NUFFT algorithm helps to reduce the computational complexity to that similar to traditional 2D FFT.

Several numerical results are used to illustrate the performance of the designed algorithm. It shows that the addressed method can obtain several orders of magnitude more accurate than traditional 2D FFT or 2D DFFT proposed in [12] under the same sampling density, as are shown in Fig. 4 and Fig. 5. To achieve the same accuracy, 2D CFT requires tens to thousands smaller sampling density and computation time than the traditional FFT algorithm, as are shown in Table 1 and Table 2.

The proposed algorithm is very useful for many applications, where the Fourier transform of discontinuous functions in an arbitrary shape boundary area is required to be computed, such as in antennas, scattering, computational electromagnetics, signal and image processing. For example, when solving the volume integral equations in electromagnetics through the well-known conjugate-gradient fast Fourier transform method, the computation of the Fourier transform for the electric current density are needed. The electric current density in application is usually discontinuous across the material interfaces and the material interfaces in general have an arbitrary shapes.

ACKNOWLEDGMENT

CHZ, YS, and LJL are partly supported by Research Fund for the Doctoral Program of Higher Education of China (RFDP) under Grants 20092302110037 and 20102302110033.

REFERENCES

1. Liu, Y. H., Z. P. Nie, and Q. H. Liu, "DIFFT: a fast and accurate algorithm for Fourier transform integrals of discontinuous functions," *IEEE Microwave and Wireless Components Letters*, Vol. 18, No. 2, 716–718, 2008.
2. Fan, Z., R. S. Chen, H. Chen, and D. Z. Ding, "Weak form nonuniform fast Fourier transform method for solving volume integral equations," *Progress In Electromagnetics Research*, Vol. 89, 275–289, 2009.
3. Xiao, K., F. Zhao, S.-L. Chai, J.-J. Mao, and L.-W. Li, "Scattering analysis of periodic arrays using combined CBF/P-FFT method," *Progress In Electromagnetics Research*, Vol. 115, 131–146, 2011.
4. Rui, X., J. Hu, and Q. H. Liu, "Fast inhomogeneous plane wave algorithm for homogeneous dielectric body of revolution," *Commun. Comput. Phys*, Vol. 8, No. 4, 917–932, 2010.
5. Yang, S., Y. Chen, and Z.-P. Nie, "Simulation of time modulated linear antenna arrays using the FDTD method," *Progress In Electromagnetics Research*, Vol. 98, 175–190, 2009.
6. Laviada-Martinez, J., Y. Alvarez Lopez, and F. Las-Heras Andrés, "Efficient determination of the near-field in the vicinity of an antenna for the estimation of its safety perimeter," *Progress In Electromagnetics Research*, Vol. 103, 371–391, 2010.
7. Huang, Y., Y. Liu, Q. H. Liu, and J. Zhang, "Improved 3-D GPR detection by NUFFT combined with MPD method," *Progress In Electromagnetics Research*, Vol. 103, 185–199, 2010.
8. Najjar-Khatirkolaei, B. N. and A. R. Sebak, "Slot antenna on a conducting elliptic cylinder coated by nonconfocal chiral media," *Progress In Electromagnetics Research*, Vol. 93, 125–143, 2009.
9. Liu, Y., Z. Liang, and Z. Yang, "Computation of electromagnetic dosimetry for human body using parallel FDTD algorithm combined with interpolation technique," *Progress In Electromagnetics Research*, Vol. 82, 95–107, 2008.
10. Swillam, M. A., M. H. Bakr, and X. Li, "Full wave sensitivity analysis of guided wave structures using FDTD," *Journal of*

Electromagnetic Waves and Applications, Vol. 22, No. 16, 2135–2145, 2008.

11. Fan, G.-X. and Q. H. Liu, "Fast Fourier transform for discontinuous functions," *IEEE Transactions on Antennas and Propagation*, Vol. 52, No. 2, 461–465, 2004.
12. Liu, Y. H., Q. H. Liu, Z. P. Nie, and Z. Q. Zhao, "Discontinuous fast Fourier transform with triangle mesh for two-dimensional discontinuous functions," *Journal of Electromagnetic Waves and Applications*, Vol. 25, No. 7, 1045–1057, 2011.
13. Froeyen, M. and L. Hellemans, "Improved algorithm for the discrete Fourier transform," *Review of ScientiFIc Instruments*, Vol. 56, 2325, 1985.
14. Sorets, E., "Fast Fourier transforms of piecewise constant functions," *Journal of Computational Physics*, Vol. 116, 369–379, 1995.
15. Zeng, P., "High-accuracy formula for discrete calculation of Fourier transforms," *Applied Mathematics and Computation*, Vol. 106, No. 2–3, 117–140, 1999.
16. Liu, Y. H., Q. H. Liu, and Z. P. Nie, "A new effcient FDTD time-to-frequency-domain conversion algorithm," *Progress In Electromagnetics Research*, Vol. 92, 33–46, 2009.
17. Zhu, C. H., Q. H. Liu, Y. Shen, and L. Liu, "A high accuracy conformal method for evaluating the discontinuous Fourier transform," *Progress In Electromagnetics Research*, Vol. 109, 425–440, 2010.
18. Liu, Q. H. and N. Nguyen, "An accurate algorithm for nonuniform fast Fourier transforms (NUFFT's)," *IEEE Microwave and Guided Wave Letters*, Vol. 8, No. 1, 18–20, 1998.
19. Nguyen, N. and Q. H. Liu, "The regular Fourier matrices and nonuniform fast Fourier transforms," *SIAM Journal on Scientific Computing*, Vol. 21, 283, 1999.
20. Song, J., Y. Liu, S. L. Gewalt, G. Cofer, G. A. Johnson, and Q. H. Liu, "Least-square NUFFT methods applied to 2-D and 3-D radially encoded MR image reconstruction," *IEEE Transactions on Biomedical Engineering*, Vol. 56, No. 4, 1134–1142, 2009.
21. Ayari, M., T. Aguili, and H. Baudrand, "New version of TWA using two-dimensional non-uniform fast Fourier mode transform (2D-NUFFMT) for full-wave investigation of microwave integrated circuits," *Progress In Electromagnetics Research B*, Vol. 15, 375–400, 2009.

22. Silvester, P. P. and R. L. Ferrari, *Finite Elements for Electrical Engineers*, Cambridge Univ. Press, 1996.

23. Notaros, B. M., "Higher order frequency-domain computational electromagnetics," *IEEE Transactions on Antennas and Propagation*, Vol. 56, No. 8, 2251–2276, 2008.

24. Cowper, G. R., "Gaussian quadrature formulas for triangles," *International Journal for Numerical Methods in Engineering*, Vol. 7, No. 3, 405–408, 1973.